2do Parcial

Algoritmos y Estructuras de Datos 3 – DC, FCEyN, UBA

18/06/2021

Para realizar consultas, deben conectarse por Discord al canal de le docente a le cual quieran consultar. Tener en cuenta que une docente no puede conectarse con dos estudiantes en simultáneo. Las aclaraciones de enunciado que podamos llegar a hacer van a ser comunicadas vía Discord al canal de consultas de la práctica.

El examen transcurre de 17:00 a 21:00 hs. A las 21:00 se desconectarán les docentes de sus canales de Discord y tendrán hasta las 21:30 para realizar la entrega vía campus. El archivo subido al campus puede sobreescribirse una cantidad ilimitada de veces hasta la hora de entrega. Independientemente de si sobreescriben o no, deberán confirmar su entrega definitiva (que ya no podrá sobreescribirse). Sólo en caso de que el Campus estuviera saturado y no funcionara, sería adecuado realizar la entrega por mail a algo3-doc@dc.uba.ar con copia a fsoulign@dc.uba.ar indicando claramente la entrega en el asunto.

El examen debe **realizarse a mano**. Deben **numerar sus hojas** y escribir en ellas sus **nombres y número de legajo** (i.e., libreta o DNI). Al finalizar, deben **escanearlo o fotografiarlo** y deben **unir y comprimir** las páginas resultantes para generar un único archivo en **formato PDF** con un **peso razonable**. El resultado debe ser un documento **legible** (buena iluminación, buena resolución, buena orientación, no fotos cortadas, etc.), **¡verificarlo!**. El archivo debe estar nombrado `apellido_nombre.extensión` y debe haber un **orden de lectura claro**.

El examen es personal y pueden usar las teóricas, las clases prácticas y las guías de ejercicios, citando claramente. Las respuestas deben estar debidamente justificadas incluso en aquellos ejercicios en los que este hecho no es recordado.

El examen se **aprueba** con al menos 2 ejercicios aprobados y al menos uno de los ejercicios 1 y 2 aprobados.

1) Un *grafo mixto* es una tripla $G = (V, E, A)$ tal que (V, E) es un grafo, (V, A) es un grafo orientado y E y A no tienen aristas en común. (En otras palabras, G se obtiene del grafo $(G, E \cup A)$ orientando las aristas de A.) Dado un grafo mixto G y un natural k, el problema de *salidas acotadas* consiste en orientar las aristas de E para obtener un digrafo D en el que el grado de salida máximo sea menor o igual a k. En caso que ningún D exista, se debe informar este hecho.

   a) Proponer un modelo de flujo que permita resolver el problema de salidas acotadas cuando un grafo mixto G y un natural k son dados.

   b) Dar una interpretación a cada unidad de flujo y cada restricción de capacidad.

   c) Indicar cómo se interpreta el flujo máximo del modelo y cómo se construye el digrafo D en caso que exista.

   d) Determinar la complejidad de resolver el modelo resultante con el algoritmo de Edmonds y Karp. La cota debe estar expresada en función de $n + m$ y debe ser lo suficientemente ajustada.

2) Considerar los siguientes problemas, donde los pesos son positivos:

- TSP: dado un digrafo completo y pesado G con vértices s y t y un natural p, ¿existe un camino de s a t que pase por todos los vertices y cuyo peso sea menor o igual a p?
- LEN-MIN-PATH: dado un digrafo completo y pesado G con vértices s y t, un natural k y un natural p, ¿existe un camino de s a t de longitud k cuyo peso sea menor o igual a p?
- k-LEN-MIN-PATH: dado un digrafo completo y pesado G con vértices s y t y un natural p, ¿existe un camino de s a t de longitud k cuyo peso sea menor o igual a p?
- k-MISS-MIN-PATH: dado un digrafo completo y pesado G con vértices s y t y un natural p, ¿existe un camino de s a t de longitud $|V(G)| - k$ cuyo peso sea menor o igual a p?

Sabiendo que TSP es NP-completo:

a) Proponer certificados y verificadores que demuestren que los otros tres problemas pertenecen a NP.[1]

b) Describir brevemente un algoritmo polinomial para k-LEN-MIN-PATH.

c) Demostrar que LEN-MIN-PATH es NP-completo. **Ayuda:** piense cómo codificar TSP como instancia de LEN-MIN-PATH.

d) Demostrar que k-MISS-MIN-PATH es NP-completo. **Ayuda:** piense cómo agregar vértices a una instancia de TSP que no puedan ser visitados.

3) Para este ejercicio, recordar que TSP es NP-completo y suponer que los resultados del ejercicio anterior son válidos.

a) Demostrar la existencia una reducción polinomial que transforma instancias de LEN-MIN-PATH en instancias de TSP.

b) Suponiendo que se descubre un algorito polinomial A para k-MISS-MIN-PATH, describir un algoritmo polinomial para TSP.

c) Demostrar que todo problema en P puede reducirse a k-LEN-MIN-PATH en tiempo polinomial, cualquiera sea el k.

4) Formalizar (i.e., describir cómo se representan las soluciones válidas, parciales y la función de extensión) un algoritmo de *backtracking* para LEN-MIN-PATH que se base en la siguientes premisas:

- Se generan subcaminos considerando únicamente los vértices que dichos caminos visitan.
- Por cada subcamino se mantiene la información de cuáles vértices ya fueron visitados para no volver a pasar por ellos.
- El algoritmo tiene una poda por optimalidad.

[1] Los tres certificados y verificadores son muy parecidos (y generalizables a un único algoritmo): no pensar que hay trampa.